



Laboratorio N° 3
Programación en lenguaje C

IMPORTANTE: lea este documento detenidamente para no cometer errores o malos entendidos en su contenido. **APAGUE** el celular. **NO SE PERMITE** el uso de sistemas de mensajería instantánea. **SI SE PERMITE** el uso de Internet, ChatGPT, Gemini, etc.

Introducción

Los punteros en C son una característica fundamental del lenguaje que permite manipular y gestionar la memoria directamente. Un puntero es una variable que almacena una dirección de memoria, en lugar de su valor. Esta capacidad otorga gran flexibilidad al programador, ya que permite optimizar el uso de la memoria, crear estructuras de datos dinámicas y manejar funciones de manera más eficiente.

Enunciado

En este laboratorio nos enfocaremos en la programación con punteros; resolver los siguientes planteos:

- a) El programa ubicado en la página 2 permite ordenar la secuencia de números pasada como parámetro a la función: `enigma()`. Encontrar cuál o cuáles son los errores que impiden alcanzar el objetivo propuesto.
- b) Una vez corregido el programa, ¿qué método de ordenamiento implementa el programa? Hacer una traza completa del programa corregido y mostrar la salida que se obtiene por pantalla. Describir brevemente el funcionamiento de cada función y del programa en general.

Para realizar este laboratorio se deberá descargar de la Plataforma Moodle el archivo **Lab03.zip** que contiene el archivo `main.c` con la implementación desarrollada en el inciso a).

```

#include <stdio.h>

int arr[10]={3,6,1,2,3,8,4,1,7,2};

int condicion(void *m, void *n){
    int *m1, *n1;
    m1 = (int *) m;
    n1 = (int *) n;

    return (*m1 > *n1);
}

void enigma(int *p, int N) {
    int i, j, t;

    for (i = N-1; i >= 0; i--)
        for (j = 1; j <= i; j++)
            if (condicion((void *) &p[j-1], (void *) &p[j] )) {
                t = p[j-1];
                p[j] = p[j-1];
                p[j-1] = t;
            } // end if
    } // end enigma

int main(void) { // el programa no usa parámetros en la línea de comando
    int i;
    putchar('\n');
    for (i = 0; i < 10; i++)
        printf("%d ", arr[i]);

    enigma(arr, 10);

    putchar('\n');
    for (i = 0; i < 10; i++)
        printf("%d ", arr[i]);

    return 0;
}

```

Sobre la implementación

- El archivo fuente principal se debe denominar **ordenar.c**.
- La compilación debe realizarse con el *flag* `-Wall` habilitado. El código debe compilar **sin advertencias** de ningún tipo.
- La copia o plagio del laboratorio es una falta grave. Quien incurra en estos actos de deshonestidad académica desaprobará el laboratorio y **el cursado de la materia**.

Estilo de programación

- El código implementado debe reflejar la aplicación de las técnicas de programación modular estudiadas a lo largo de la carrera.
- En el código, entre eficiencia y claridad, se debe optar por la claridad. Toda decisión en este sentido, de hacer falta, debe constar en la documentación interna del código fuente asociado al programa implementado.
- El código debe estar indentado, comentado y debe reflejar el uso adecuado de nombres significativos para la definición de variables, funciones y parámetros.

Entrega

- La entrega del código fuente se realizará a través de un archivo comprimido **zip** o **rar**, denominado **LAB3-Comision-XX** (el valor XX debe modificarse por el número de comisión correspondiente) donde se deben incorporar el/los archivos fuente “.c” y “.h”.
- El archivo comprimido debe subirse a la plataforma Moodle, en el apartado *Laboratorios de programación*, dentro de la tarea Laboratorio 3 [actividad]. Para esto, tenga en cuenta las siguientes consideraciones:
 - **Fecha de entrega: martes 22/10/24, 14:00 hs.**
Considere que la plataforma Moodle de forma **automática inhabilitará** la opción de entrega a partir de esta hora, por lo que se recomienda no esperar a último momento para realizar la entrega.
 - **Intentos de entrega:** la plataforma Moodle **no permitirá** realizar más de una entrega de los archivos. No se aceptará ninguna entrega fuera de la plataforma Moodle.
 - El archivo comprimido debe subirlo **sólo uno** de los integrantes de la comisión.
 - La cátedra no se responsabiliza por cualquier inconveniente surgido a partir del envío del proyecto a último momento. Es importante **no esperar hasta último momento** para realizar el envío evitando así todo tipo de situación excepcional que pudiese ocurrir (sobrecarga o caída del sistema, fallas en la conexión a Internet, etc.).

Corrección

- La cátedra evaluará tanto el **diseño, implementación, presentación** y cumplimiento de **todas** las condiciones de entrega.